

### **AMENDMENTS TO THE CLAIMS**

This listing of the claims will replace all prior versions and listings of claims in the application.

**Cancel claims 1- 6, previously restricted from the application and withdrawn, without prejudice.**

1. (Cancelled)
2. (Cancelled)
3. (Cancelled)
4. (Cancelled)
5. (Cancelled)
6. (Cancelled)

7. (Currently Amended) A method of loading an application program written to operate only on a single computer onto each of a plurality of computers, the plurality of computers being interconnectable via a communications link without forming a distributed shared memory arrangement, and different portions of said application program being simultaneously executable on each different one of the plurality of computers with each different one of the plurality of computers having a different independent local memory accessible only by a corresponding portion of the application program, the method comprising the steps of:

loading the application program written to operate only on a single computer onto each different computer of said plurality of computers; and

modifying the application program on each said different computer before execution of said corresponding portion of the application program written to operate only on a single computer on each said different computer.

8. (Previously Presented) The method as claimed in claim 7 wherein the step of modifying the application program is different for different computers.

9. (Previously Presented) The method as claimed in claim 7 or 8 wherein said modifying step comprises:

- (i) detecting instructions which share memory records,

(ii) listing all such shared memory records and providing a naming tag for each listed memory record,

(iii) detecting those instructions which write to, or manipulate the contents of, any of said listed memory records, and

(iv) generating an alert instruction corresponding to each said detected write or manipulate instruction, said alert instruction forwarding the re-written or manipulated contents and name tag of each said re-written or manipulated listed memory record.

10. (Canceled).

**Cancel claims 11- 13, previously restricted from the application and withdrawn, without prejudice.**

11. (Cancelled)

12. (Cancelled)

13. (Cancelled)

14. (Currently Amended) A method of compiling or modifying an application program written to operate on only a single computer but to run simultaneously on each one of a plurality of computers ~~interconnectable~~interconnectable via a communications link without forming a distributed shared memory arrangement, with different portions of said application program being simultaneously executable on different ones of said plurality of computers with each one of the plurality of computers having an independent local memory accessible only by the corresponding portion of the application program, said method comprising the steps of:

(i) detecting instructions which share memory records,

(ii) listing all such shared memory records and providing a naming tag for each listed memory record,

(iii) detecting those instructions which write to, or manipulate the contents of, any of said listed memory records, and

(iv) generating an alert instruction following each said detected write or manipulate instruction, said alert instruction forwarding the re-written or manipulated contents and name tag of each said re-written or manipulated listed memory record.

15. (Original) The method as claimed in claim 14 and carried out prior to loading the application program onto each said computer.

16. (Original) The method as claimed in claim 14 and carried out during loading of the application program onto each said computer.

17. (Original) The method as claimed in claim 14 and carried out by just-in-time compilation.

18. (Original) The method as claimed in claim 14 and carried out by re-compilation after loading .

19. (Canceled).

**Cancel claims 20-23, previously restricted from the application and withdrawn, without prejudice.**

20. (Cancelled)

21. (Cancelled)

22. (Cancelled)

23. (Cancelled)

24. (Previously Presented) A method of loading an application program as in claim 7, wherein said program written to operate on only a single computer is a program written to execute within a local processor or processors and local memory coupled to the processor or processors within the single computer.

25. (Currently Amended) A method of loading an application program as in claim 7, wherein each of the computers operates with the same application program and data and thus all of the plurality of computers have the same application program and data.

26. (Previously Presented) A method of compiling or modifying an application program as in claim 14, wherein said program written to operate on only a single computer is a program

written to execute within a local processor or processors and local memory coupled to the processor or processors within the single computer.

27. (Currently Amended) A method of loading an application program written to execute on only a single computer onto each one of a plurality of computers interconnectable via a communications link to enable simultaneous cooperative execution of said application program, with different portions of said application program being simultaneously executed within a different independent local processor and a different independent local memory within each different one of the plurality of computers, said different independent local memory within each said different computer not forming a distributed shared memory arrangement, and being accessible during execution of said application program and said different portions of said application program only by the different portion of the application program actually executing within the different computer, the method comprising the steps of:

loading the application program onto each different computer of said plurality of computers;  
and

modifying the application program on each said different computer before execution of said different portion of the application program on each said different computer.

28. (Currently Amended) A method of compiling or modifying an application program written to execute on only a single computer to run simultaneously on each one of a plurality of computers ~~interconnectable~~ interconnectable via a communications link without forming a distributed shared memory arrangement, with different portions of said application program being simultaneously executable within a different independent local processor and a different independent local memory within each different one of the plurality of computers, said different independent local memory within each said different computer being accessible during execution of said application program and said different portions of said application program only by the different portion of the application program actually executing within the different computer, said method comprising the steps of:

- (i) detecting instructions which share memory records,
- (ii) listing all such shared memory records and providing a naming tag for each listed memory record,

(iii) detecting those instructions which write to, or manipulate the contents of, any of said listed memory records, and

(iv) generating an alert instruction following each said detected write or manipulate instruction, said alert instruction forwarding the re-written or manipulated contents and name tag of each said re-written or manipulated listed memory record.

**Add new claims 29-30 as follows:**

29. (New) The method as claimed in claim 7, wherein the interconnection of the plurality of computers via the communications link without forming a distributed shared memory arrangement and the different portions of said application program being simultaneously executable on each different one of the plurality of computers with each different one of the plurality of computers having a different independent local memory accessible only by a corresponding portion of the application program eliminate clock cycle delays that would otherwise be associated with one or said plurality of computers reading memory physically located in a different one or ones of the plurality of computers formed in a distributed shared memory arrangement.

30. (New) The method as claimed in claim 25, wherein the interconnection of the plurality of computers via the communications link without forming a distributed shared memory arrangement and the different portions of said application program being simultaneously executable on each different one of the plurality of computers with each different one of the plurality of computers having a different independent local memory accessible only by a corresponding portion of the application program eliminate clock cycle delays that would otherwise be associated with one or said plurality of computers reading memory physically located in a different one or ones of the plurality of computers formed in a distributed shared memory arrangement.